

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method of translating data, comprising:
obtaining a value of an implementation data structure from an instrumented program,
wherein the implementation data structure is internal to the instrumented program;
accessing a translator associated with the instrumented program, wherein the translator
comprises a plurality of transformations; and
translating the value of the implementation data structure using the translator to obtain
translated data, wherein the translating comprises applying at least one of the
plurality of transformations to convert a representation of the implementation data
structure into an interface data structure, wherein the interface data structure
corresponds to an interface offered by the instrumented program;
wherein the translated data is configured to satisfy an instrumentation request from a user.
2. (Currently Amended) The method of claim 1, further comprising:
executing a tracing program to enable a probe in the instrumented program based on the
instrumentation request;
triggering the probe in the instrumented program; and
transferring translated data from the translator to an execution framework, wherein the
execution framework comprises a tracing framework.
3. (Currently Amended) The method of claim 1, further comprising:
executing a debugging program in the instrumented program in response to [[an]] the
instrumentation request; and
transferring translated data to an execution framework in response to the ~~instrumentation~~
instrumentation request, wherein the execution framework comprises a debugger.
4. (Original) The method of claim 1, wherein the translator is defined using a high-level
programming language.

5. (Original) The method of claim 1, wherein the translator is updated independently of the execution framework.
6. (Original) The method of claim 1, further comprising:
delivering the translator using an encoded delivery.
7. (Original) The method of claim 1, further comprising:
delivering the translator using a compiled delivery.
8. (Original) The method of claim 1, further comprising:
selecting the translator using an instrumentation request.
9. (Original) The method of claim 1, further comprising:
selecting the translator using knowledge of a function argument type of the instrumented program.
10. (Currently Amended) A system for translating data, comprising:
a memory configured to:
 store an instrumented program comprising at least one implementation data structure,
 wherein the implementation data structure is internal to the instrumented
 program;
 store a translator comprising a plurality of transformations;
a processor configured to:
 execute a compiler arranged to compile the plurality of transformations into the
 translator accept the translator and transform at least one implementation data
 structure into translated data; and
 execute an execution framework configured to receive the translated data use the
 translator to convert at least one implementation data structure into an
 interface data structure to obtain translated data, wherein the interface data
 structure corresponds to an interface offered by the instrumented program.

and wherein the translated data is configured to satisfy an instrumentation request from a user.

11. (Currently Amended) The system of claim 10, wherein ~~[[an]]~~ the instrumentation request explicitly translates the value of the at least one implementation data structure into the translated data.
12. (Original) The system of claim 10, wherein a function call implicitly triggers the translating the value of the at least one implementation data structure into the translated data.
13. (Original) The system of claim 10, wherein the translator is defined using a high-level programming language.
14. (Original) The system of claim 10, wherein the translator is updated independently of the execution framework.
15. (Original) The system of claim 10, wherein the translator is delivered using at least one selected from the group consisting of encoded delivery and compiled delivery.
16. (Original) The system of claim 10, wherein the execution framework comprises at least one selected from the group consisting of a tracing framework and a debugger.

17. (Currently Amended) A computer system for translating data, comprising:
a processor;
a memory;
a storage device; and
software instructions stored in the memory for enabling the computer system to:
 obtain a value of an implementation data structure from an instrumented program,
 wherein the implementation data structure is internal to the instrumented
 program;
 access a translator associated with the instrumented program, wherein the translator
 comprises a plurality of transformations; and
 translate the value of the implementation data structure using the translator to obtain
 translated data, wherein the translating comprises applying at least one of the
 plurality of transformations to convert a representation of the implementation
 data structure into an interface data structure, wherein the interface data
 structure corresponds to an interface offered by the instrumented program;
 wherein the translated data is configured to satisfy the instrumentation request from a
 user.
18. (Original) The computer system of claim 17, wherein the translator is defined using a high-level programming language.
19. (Original) The computer system of claim 17, wherein the translator is updated independently of the execution framework.
20. (Original) The computer system of claim 17, further comprising software instructions to deliver the translator using an encoded delivery.
21. (Original) The computer system of claim 17, further comprising software instructions to deliver the translator using a compiled delivery.

22. (Original) The computer system of claim 17, further comprising software instructions to select the translator using the instrumentation request.
23. (Original) The computer system of claim 17, further comprising software instructions to select the translator using knowledge of a function argument type of the instrumented program.